ISSN 2237-7166

As k-melhores soluções do problema da mochila para a resolução do problema de corte unidimensional com sobras aproveitáveis

Renata T. Fonseca*
Bacharelado em Sistemas de Informação, UNESP
17033-360, Bauru, SP
E-mail: reetiepo@gmail.com

Adriana C. Cherri
Departamento de Matemática, UNESP
17033-360, Bauru, SP
E-mail: adriana@fc.unesp.br

Andréa Vianna
Departamento de Computação, UNESP
17033-360, Bauru, SP
E-mail: vianna@fc.unesp.br

RESUMO

Nos problemas de corte de estoque unidimensional com sobras aproveitáveis (PCESA), um conjunto de itens deve ser produzido a partir do corte de um conjunto de objetos disponíveis em estoque, de modo que a perda dos padrões de corte seja mínima. Nestes problemas, como a qualidade dos padrões de corte depende diretamente dos tamanhos e quantidades dos itens a serem produzidos, pode-se considerar que se a demanda presente gerar sobras indesejáveis (nem tão grandes para serem aproveitáveis, nem tão pequenas para serem perdas aceitáveis), então convém gerar retalhos (não computáveis como perda) que podem ser utilizados para produzir itens de demandas futuras.

O aproveitamento de sobra das peças cortadas, embora citado por Brown [1], passou a ser considerado de maneira explícita em estudos recentes de problemas de corte de estoque. Scheithauer [4] modificou o problema proposto por Gilmore e Gomory [3] para resolver o PCESA. A estratégia proposta consiste em incluir itens extras aos demandados sem haver demandas para serem atendidas. Estes itens extras que podem ser ou não incluídos no padrão de corte são os possíveis retalhos para o problema. Este problema proposto por Scheithauer [4] foi modificado por Cui e Yang [2], que incluíram limitações no estoque de objetos e na quantidade de retalhos que podem ser geradas. O modelo matemático utilizado pelos autores é dado por:

$$Min\ z = \sum_{k=1}^{K}\sum_{j=1}^{N_k} c_{jk}x_{jk} - \sum_{k=1}^{K}\sum_{j=1}^{N_k}\sum_{i=m+1}^{n} w_{i-m}\,\alpha_{ijk}x_{jk} \qquad (1)$$

$$\text{Sujeito a: } \begin{cases} \sum_{k=1}^{K}\sum_{j=1}^{N_k}\alpha_{ijk}x_{jk} \geq d_i \ , i=1,\dots,m & (2)\\ \sum_{k=1}^{K}\sum_{j=1}^{N_k}\alpha_{ijk}x_{jk} \leq u_{i-m} \ , \quad i=m+1,\dots,n & (3)\\ \sum_{j=1}^{N_k} x_{jk} \leq e_k \ , \quad k=1,\dots,K & (4)\\ x_{jk} \geq 0 \text{ e inteiro } j=1,\dots,N_k \ , k=1,\dots,K & (5) \end{cases}$$

em que:

- c_{jk}: custo de cortar o objeto tipo k no padrão de corte j, $k = 1, \dots, K$, $j = 1, \dots, N_k$;
- w_{i-m}: custo de cortar o retalho i do estoque, $i = m + 1, \dots, n$;

* Bolsista de Iniciação Científica FAPESP

 ISSN 2237-7166

- α_{ijk}: número de itens do tipo i cortados no padrão de corte j para o objeto em estoque do tipo k, $i = 1, \ldots, m$, $j = 1, ..., N_k$, $k = 1, \ldots, K$;
- α_{ijk}: número de retalhos do tipo i cortados no padrão de corte j para o objeto em estoque do tipo k, $i = m + 1, \ldots, n$, $j = 1, ..., N_k$, $k = 1, \ldots, K$
- d_i: demanda do item tipo i, $i = 1, \ldots, m$;
- e_k: número de objetos tipo k em estoque, $k = 1, \ldots, K$;
- u_{i-m}: número máximo de novos retalhos, $i = m+1, \ldots, n$.
- x_{jk}: número de objetos tipo k cortados no padrão de corte j, $k = 1, \ldots, K$, $j = 1, ..., N_k$.

Para resolver o problema, os autores utilizaram a técnica de programação dinâmica, sendo que os retalhos previamente definidos (itens extras) também são considerados durante esse processo, porém, sem demandas para serem atendidas. Para obter uma solução inteira para o problema, os autores utilizaram um procedimento heurístico proposto em um trabalho anterior.

Neste trabalho utilizamos o mesmo modelo matemático proposto por Cui e Yang [2] para resolver o PCESA. Para resolver o problema da mochila que surge durante a geração de colunas, utilizamos algoritmo *Branch & Bound* com a estratégia de gerar as k-melhores soluções para o problema, sendo que os retalhos também são considerados durante esse processo. Para obter uma solução inteira a partir da solução contínua do problema, utilizamos as heurísticas RAG (Residual de Arredondamento Guloso) – versão 1 e 2 propostas por Poldi e Arenales [5].

Atualmente está sendo desenvolvido o algoritmo para a resolução das k-melhores soluções para o problema da mochila. O código foi implementado na linguagem C++, utilizando o Visual Studio 2008 em ambiente Windows 7.

Alguns testes preliminares utilizando a geração de uma única coluna foram realizados para verificar o desempenho do algoritmo implementado. Os exemplos a seguir apresentam dois tipos objetos com comprimentos padronizados de 1000 e 1100. Os comprimentos dos itens demandados foram gerados aleatoriamente no intervalo [150, 320] e as demandas no intervalo [10, 300]. As quantidades de tipos de itens m, foram definidas no intervalo [25, 40]. Os comprimentos definidos para os possíveis retalhos foram 300, 400 e 500. Outros valores poderiam ser utilizados. A tabela abaixo apresenta os resultados obtidos:

Exemplos	***Perda***	***Retalho 300***	***Retalho 400***	***Retalho 500***
1	198	1		1
2	173	2		
3	88	1	1	
4	110	1		1
5	238			

Pela tabela observamos que os exemplos não geram todos os comprimentos de retalhos. O comprimento da perda poderia ser menor se o comprimento dos retalhos não fosse padronizado (300, 400 ou 500), ou se uma maior variedade dos mesmos fosse definida. Esses comprimentos foram definidos para os retalhos devido ao intervalo no qual os comprimentos dos itens foram definidos. Como a ideia é gerar itens para atender demandas futuras consideramos que esses comprimentos seriam satisfatórios.

Referências

[1] Brown, A. R., *Optimum packing and depletion: the computer in space and resource usage problem*. New York: Macdonald – London and American Elsevier Inc, 1971.107p, (1971).

[2] Cui, Y., Yang, Y., *A heuristic for the one-dimensional cutting stock problem with usable leftover*. European Journal of Operational Research, 204: 245-250, 2010.

ISSN 2237-7166

[3] Gilmore, P. C., Gomory, R. E., *A linear programming approach to the cutting stock problem – Part II*. Operational Research, 11: 863-888, 1963.
[4] Scheithauer, G., *A note on handling residual length*. Optimization, 22: 461-466, 1991.
[5] Poldi, K. C., Arenales, M. N., *Heuristics for the one-dimensional cutting stock problem with limited multiple stock lengths*. Computer and Operations Research, 36: 2074-2081, 2009.